Theor Chem Acc (2010) 125:599–611
DOI 10.1007/s00214-009-0648-3

REGULAR ARTICLE

Optimization of multi-classifiers for computational biology: application to gene finding and expression

Rocío Romero-Zaliz · Cristina Rubio-Escudero · Igor Zwir · Coral del Val

Received: 17 March 2009 / Accepted: 23 September 2009 / Published online: 15 October 2009

Abstract Genomes of many organisms have been sequenced over the last few years. However, transforming such raw sequence data into knowledge remains a hard task. A great number of prediction programs have been developed to address part of this problem: the location of genes along a genome and their expression. We propose a multi-objective methodology to combine state-of-the-art algorithms into an aggregation scheme in order to obtain optimal methods' aggregations. The results obtained show a major improvement in sensitivity when our methodology is compared to the performance of individual methods for gene finding and gene expression problems. The methodology proposed here is an automatic method generator, and a step forward to exploit all already existing methods, by providing alternative optimal methods' aggregations to answer concrete queries for a certain biological problem with a maximized accuracy of the prediction. As more approaches are integrated for each of the presented problems, de novo accuracy can be expected to improve further.

Keywords Multiobjective · Gene finding · Gene expression

Dedicated to Professor Sandor Suhai on the occasion of his 65th birthday and published as part of the Suhai Festschrift Issue.

Rocío Romero-Zaliz, Cristina Rubio-Escudero contributed equally.

R. Romero-Zaliz · I. Zwir · C. del Val (✉)
Computer Science and Artificial Intelligence Department, Granada University, Granada, Spain
e-mail: delval@decsai.ugr.es

C. Rubio-Escudero
Departamento de Lenguajes y Sistemas Informáticos, Sevilla University, Sevilla, Spain

I. Zwir
Howard Hughes Medical Institute, Department of Molecular Microbiology, Washington University School of Medicine, St. Louis, MO, USA

1 Introduction

Genomes of many organisms have been sequenced over the last few years. However, transforming such raw sequence data into knowledge remains a hard task [1]. A great number of prediction programs have been developed to address one part of this problem: the location of genes along a genome [2–4]. Unfortunately, finding genes in a genomic sequence is far from being a trivial problem. Computational gene prediction methods yet have to achieve perfect accuracy, even in the relatively simple prokaryotic genomes [1]. Gene prediction is one of the most important problems in computational biology due to the inherent value of the set of protein-coding genes for other analyses.

Another part of the problem is determining when, where and for how long these genes are turned on or off. Microarray technology allows the simultaneous evaluation of the expression of hundreds of genes in a single assay, converting this technology in a powerful tool for expression profiling, as well as, diagnosis and classification of cancers and other diseases. However, this technology presents a wealth of analysis problems [5] such as the inherent variability of cDNA microarrays at the individual slide and spot level, the large-scale nature of the data, and the fact that the full use of expression profiles for inferring gene function is still only partly explored. Many new methods have been developed to address the statistical challenge of

identifying "important" genes in the large sets of raw sequence data [6–11]. However, there is still a dearth of computational methods to facilitate understanding of differential gene expression profiles (e.g., profiles that change over time and/or over treatments and/or over patient) and to decide which of the many available statistical methods is the most reliable to identify differences across profiles.

Despite the advances in both referred problems, existing approaches to predict genes and to analyze microarray data have intrinsic advantages and limitations [1, 12]. Furthermore, there is no program or methodology that can provide perfect predictions for any given input for either of these two problems.

The problems of gene finding (identifying genes, exons and introns, beginning and end of the genes) and analysis of gene expression are formulated in this paper as classification problems. The gene finding problem can be interpreted as a simple decision between which section of a sequence is protein coding and which is not. Concerning the gene expression from microarray experiments, the classification problem can be seen as a decision between which genes are active or inactive in a given time point and/or under a given condition. For both problems many different programs are available, which give distinct solutions. There have been previous approaches to combine gene predictors [13–15] and microarray analyses [16, 17], but maximizing accuracy by weighting both sensitivity and specificity functions into a single objective. However, our methodology uses a multi-objective approach to extract the best methods' aggregations by maximizing the specificity and sensitivity of their predictions individually. This approach combines state-of-the-art algorithms into an aggregation scheme to provide better predictions by taking advantage of the different methodologies' strengths and avoiding their weaknesses.

We applied our methodology to the both referred problems. In the gene finding problem, we used the EGASP sets from the ENCODE Genome Annotation Assessment Project (EGASP) [18, 19]. These datasets contain manually curated fragments of the human genome originating from the ENCODE project [20]. This data set was selected by the EGASP assessment because the genes encoded in these regions were not used to train any particular gene predictor. Therefore, it is not a biased dataset. In the case of analysis of the microarrays, we used a dataset derived from the analysis of longitudinal blood expression profiles of human volunteers treated with intravenous endotoxin, compared to those treated with a placebo in order to study the inflammation and human response to injury. This dataset was part of a Large-scale Collaborative Research Project sponsored by the National Institute of General Medical Sciences [21].

2 Materials and methods

The aggregation of the results from various methods is accomplished using the union -∪- and intersection -∩- operators [22]. All potential aggregations, termed methods' aggregations from here on, form a space of potential hypotheses, which can be represented as a lattice structure (Fig. 1). We search for the best methods' aggregations, moving from hypothesis to hypothesis towards the most general, the union of all methods, and the most specific, their intersection, which are located at the top and the bottom of the lattice, respectively [23] (Fig. 1). In Fig. 1 we can appreciate the lattice generated by the union and intersection of methods. In the gene finding problem we explore five methods, $n = 5$, termed M_1 to M_5, forming a total set of 31 potential aggregations. In the analysis of microarrays problem, ten methods are surveyed, $n = 10$, termed M_1 to M_{10}, forming a total set of 1,023 potential aggregations. The methods' aggregations are evaluated based on a multi-objective approach [24] to extract the best methods' aggregations by maximizing the specificity (Sp) and sensitivity (Sn) of their predictions. To estimate the sampling bias [25] of the methods' aggregations we randomly partition the original sample into ten subsamples, and each subsample was retained as a validation data for each methods' aggregations.

2.1 Gene finding problem: dataset and programs

For the gene finding problem, we selected 27 ENCODE regions to test our proposal. These ENCODE regions have undergone an exhaustive annotation strategy prior to EGASP by the HAVANA team [26]. They consist of 2,471 total transcripts representing 434 unique protein-coding gene loci.

The programs used in this study are those used in the EGASP competition, which are ab initio gene predictors using a single genome sequence. These programs were designed to predict gene structure, or at least a set of spliceable exons in vertebrate or pre-human genome sequences: GeneID [27], Genscan [28], Genemark [29], Augustus [30] and GeneZilla [31]. GeneID combines different algorithms using Position Weight Arrays to detect features such as splice sites, start and stop codons and Markov Models to score exons and Dynamic Programming (DP) to assemble the gene structure [27]. Genescan uses a general probabilistic model for the gene structure of human genomic sequences. It has the capacity to predict multiple genes in a sequence, to deal with partial as well as complete genes, and to predict consistent sets of genes occurring on either or both DNA strands [32]. Genemark for eukaryotes gathers the original Genemark models into the naturally derived hidden Markov model framework with

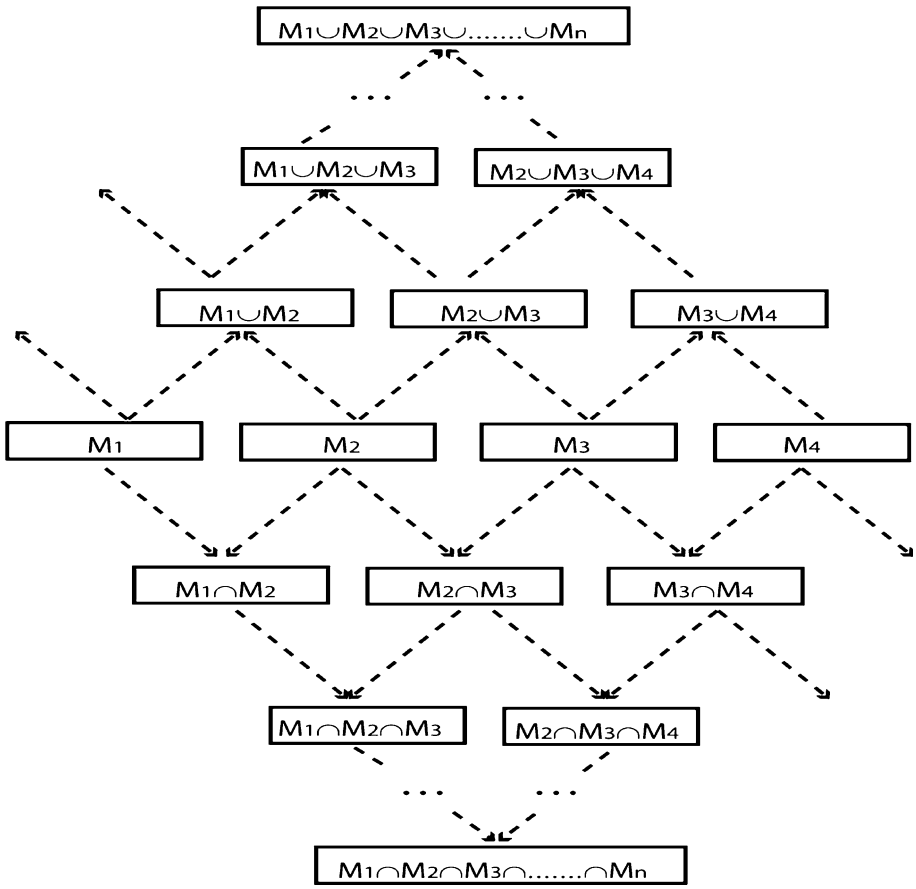

Fig. 1 Lattice of potential hypothesis, methods' aggregations of $M_1 \ldots M_n$ using the -∪- and -∩- and operators. The *solid arrows* show the direction of the search in the space of hypotheses

gene boundaries modeled as transitions between hidden states [29]. Augustus is a gene predictor for eukaryotic genomic sequences that is based on a generalized hidden Markov model, a probabilistic model of a sequence and its gene structure [30]. GeneZilla is based on the Generalized Hidden Markov Model (GHMM) framework, similar to Genscan. Graph-theoretic representations of the high scoring open reading frames are provided, allowing for exploration of sub-optimal gene models. It makes use of Interpolated Markov Models (IMMs), Maximal Dependence Decomposition (MDD), and includes states for signal peptides, branch points, TATA boxes and CAP sites [31]. For each method, the closest organism available for each gene in the dataset was selected. Predictions on both strands were extracted.

The aggregations of the results of the different gene prediction approaches are performed at a nucleotide level. The aggregation of the results of different methods joins two or more overlapping or adjacent exons into a larger new exon (Fig. 2). Nucleotide level accuracy is calculated as a comparison of the annotated nucleotides with the predicted nucleotides. Individual nucleotides appearing in more than one transcript in either the annotation or the predictions are considered only once for the nucleotide level statistics. Nucleotide predictions must be on the same strand as the annotations to be counted as correct. At the nucleotide level, Sn is the proportion of annotated nucleotides (as being coding or part of an mRNA molecule) that is correctly predicted, and Sp the proportion of predicted nucleotides (as being coding or part of an mRNA molecule) that is so annotated. As a summary measure, we have computed the correlation coefficient between the annotated and the predicted nucleotides [19, 33].

$$\mathrm{Sp} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \quad \mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{1}$$

$$\mathrm{CC} = \frac{(\mathrm{TP}\times\mathrm{TN}) - (\mathrm{FN}\times\mathrm{FP})}{\sqrt{(\mathrm{TP}+\mathrm{FN})\times(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})}} \tag{2}$$

2.2 Gene expression profile finding: datasets and analysis methods

The dataset used was derived from longitudinal blood expression profiles of human volunteers treated with intravenous endotoxin compared to those treated with a placebo. The data are related to the host response over time to systemic inflammatory insults, as part of a large-scale collaborative research project sponsored by the National Institute

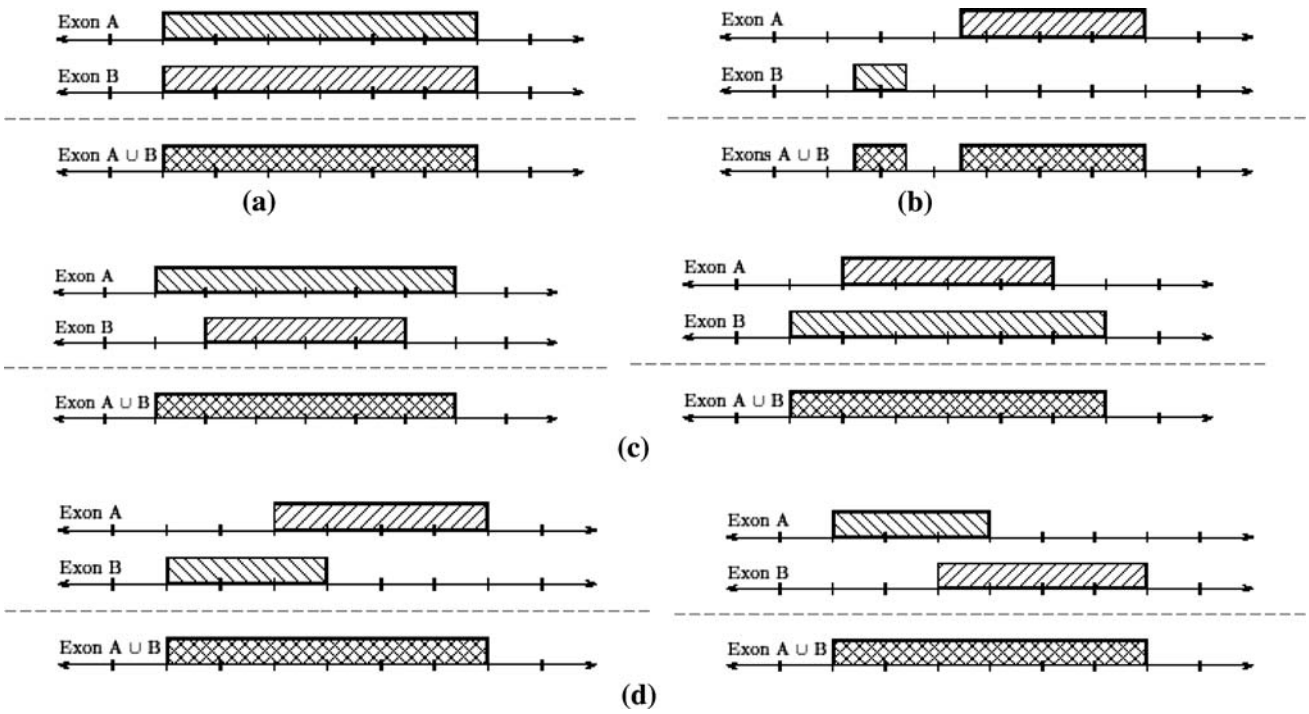

Fig. 2 Aggregation of exons using the union operator **a** equal exons, **b** missed exons, **c** included exons, **d** overlapped exons

of General Medical Sciences (http://www.gluegrant.org). The data was derived from blood samples collected from eight normal human volunteers, four treated with intravenous endotoxin (i.e., patients 1–4) and four with placebo (i.e., patients 5–8) [21]. Complementary RNA was generated from circulating leukocytes at 0, 2, 4, 6, 9 and 24 h after the intravenous infusion and hybridized with GeneChips® HG-U133A v2.0 from Affymetryx Inc., containing a set of 22,283 probe sets. A total set of 29 gene expression profiles (sets of genes which exhibit a common behavior throughout the conditions of the problem under study, time, treatment and patient in our particular case) are contained in the dataset and forms the focus of our study [34].

The methods analyzed in this study were applied to identify meaningful gene expression profiles from microarray data. The list of programs used comprises the methods most frequently applied to the analysis of microarray data: Student's *t* test [7], Permutation Test [10], Analysis of Variance (ANOVA) [7] and Repeated Measures Analysis of Variance (RMANOVA) [35]. These methods have been applied to the inflammation and host response to injury problem to account for different experimental conditions, such as treatment versus control and different time points. Therefore, Student's *t* test and Permutation Test have been applied in two different ways: considering treatment versus control and considering time. The ANOVA and RMANOVA tests can account for more than one experimental condition simultaneously; therefore they have been applied in three different ways: considering treatment versus control, considering time, and considering treatment versus control and time simultaneously.

The aggregation of the results of different methods in the Gene Expression Profile Finding Problem is performed by combining the results obtained by each of the individual methods (group of probe sets). The union of two methods, M_a and M_b ($M_a \cup M_b$), is defined as the group resulting which contains all genes retrieved either by method M_a or by M_b. The intersection of two methods, M_a and M_b ($M_a \cap M_b$), is defined as the group resulting which contains all genes retrieved by both methods M_a and M_b.

We evaluate the performance of each method aggregation to retrieve each of the 29 gene expression profiles present in the inflammation and host response to injury dataset. In our particular problem, when studying the behavior of method M_i to retrieve a gene expression profile P_j, we define true positives (TPs) as probe sets retrieved by method M_i which exhibit the gene expression profile P_j, true negatives (TNs) probe sets not retrieved by method M_i which do not exhibit the gene expression profile P_j, false positives (FPs) as probe sets retrieved by method M_i which do not exhibit the gene expression profile P_j, and false negatives (FNs) as probe sets not retrieved by method M_i which exhibit the gene expression profile P_j.

TP, TN, FP and FN information is typically summarized in terms of Sn, the proportion of probe sets belonging to P_j in the dataset and correctly retrieved by the method M_i under evaluation, and Sp, the proportion of probe sets correctly retrieved by the method M_i from all the probe sets

retrieved by method M_i (see Eq. 1). These measures are formally described as for the Gene Finding Problem.

3 Results

The results obtained applying our methodology to the two proposed biological problems outperform in terms of specificity and sensitivity the results obtained by classical methods, though both gene prediction and the identification of gene expression profiles are problems of different nature.

3.1 Gene finding

The most updated version of the ENCODE annotations and of each gene finding prediction algorithm was used. The specificity, sensitivity and correlation coefficient (CC) averages for each individual method are shown in Table 1 with values represented in the [0–1] interval (see Eqs. 1, 2).

Genscan showed the highest CC while GeneMark obtained the lowest CC. GeneID obtained the highest specificity and the lowest sensitivity, while GeneZilla showed the highest sensitivity with lower specificity. The analysis of the individual results shows that some algorithms are able to predict certain genes very accurately with CC values close to 1, but the same algorithm completely fails to predict other genes (CC below 0.7 or even 0.5). These results show that a high average CC does not imply a good performance, and vice versa, since the average might hide some low CCs for specific genes.

We evaluated all possible 31 methods' aggregations resulting from applying the union and intersection operators to the selected five gene finding programs. The results (Fig. 3) show the general increase of sensitivity and the decrease of specificity with the increasing number of methods per aggregation when applying the union operator. However, we find the opposite behavior when using the intersection operator.

The comparison between the prediction accuracy of the individual methods and the aggregation strategy can be seen in Fig. 3. This figure shows that methods' aggregations increase either prediction's sensitivity or specificity, and a few aggregations increase both objectives when compared with the individual methods. The best results are obtained in aggregations containing 2 or 3 methods via either union or intersection. Figure 4 shows the Pareto optimal front [36] in red, which is the best methods' aggregation in terms of sensitivity and specificity simultaneously. The Pareto optimal front consists of those methods' aggregations for which improvement in specificity can only occur with the worsening of sensitivity and vice versa, that is, the best method aggregations in terms of sensitivity and specificity, simultaneously. Individual methods are represented in blue.

The top ten methods' aggregations are shown in Table 2. The results show that aggregations improving the individual methods' performances in sensitivity and specificity generally include Augustus. There are several best methods' aggregations, including union and intersection, but the one requiring the lower number of methods and predicting the highest number of genes is the union Augustus ∪ GeneID. The best combination is Genscan ∩ GeneZilla when considering the intersection operator. The former correctly predicts 80.00% of the dataset, while the latter achieves 73.33%. GeneZilla is present in many of the best ten methods' aggregations, providing supplementary predictions to the other methods, even though it is the worst individual method (Table 1).

Table 1 Individual gene finding method's performance

Method	Specificity	Sensitivity	Correlation coefficient
Genscan	0.772	0.759	**0.745**
Genzilla	0.750	**0.782**	0.744
Augustus	0.794	0.694	0.724
GeneID	**0.829**	0.658	0.712
GeneMark	0.764	0.664	0.683

The methods are ordered according to their correlation coefficient; the best result for each column is highlighted in bold and the worst in italic

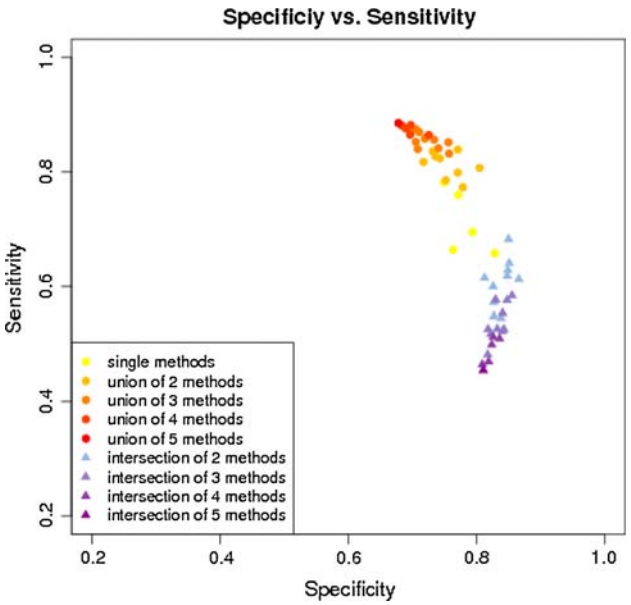

Fig. 3 Sensitivity versus specificity for all methods' aggregations and individual methods. The number of methods used in each aggregation is color-coded as indicated in the legend. The five *light-yellow circles* represent sensitivity and specificity values for the individual methods

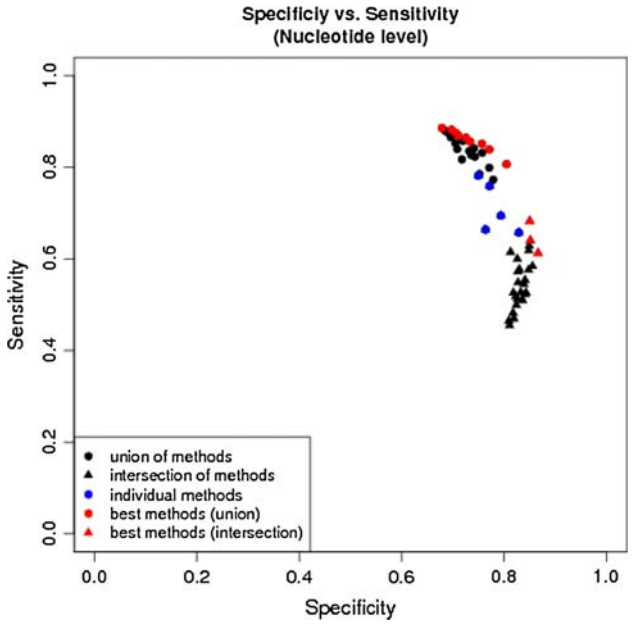

Fig. 4 Sensitivity versus specificity: Optimal Pareto front. There are three color-codes: *blue dots* correspond to single methods, *red dots* to those methods' aggregations that are optimal in both objectives sensitivity and specificity (Pareto optimal front); and *black dots* are all other methods' aggregations

Cross-validation techniques are often used to estimate how accurately a predictive model will perform in practice, and consequently, to specifically avoid overfitting [23]. Unfortunately, because the original training datasets and some programmed code of the individual methods are not accessible, they cannot be re-trained based on different data partitions and tested with the remaining ones. Therefore, to indirectly estimate the sampling bias of the methods' aggregations [25], we randomly partition the original sample into ten subsamples, and each subsample was used as validation dataset for each methods' aggregation (Table 2). The results obtained show a small sampling variability (Fig. 5), suggesting good performance and robust results for the methods' aggregations selected as the best ones.

The levels of specificity and sensitivity obtained by each methods' aggregation over the complete dataset (ENCODE) are shown in Fig. 6, ranging from 0 (green) to 1 (red). Each row represents a methods' aggregation and each column a gene from the dataset. The rows and columns for each graph were clustered independently, and therefore we can see groups (clusters) of method's aggregations showing similar behavior. For instance, in Fig. 6a, there are several methods' aggregations using the union operator that cannot predict some ENCODE genes; those are represented as green columns (e.g., AC068580, AC079630, AC021607). Regions, which are easy to detect by many of the methods' aggregations are represented as mostly red columns (e.g., AC093511, AC131574, AC113331).

The sensitivity of each method aggregation using the union operator to predict each gene is shown in Fig. 6b. There are a few genes (e.g., AC068580, AC079630, AC021607) that obtain very low sensitivity for all methods' aggregations, as it can be seen in their green columns. The aggregation of methods increases the sensitivity of the prediction as it is shown by red cells in Fig. 6b.

Figure 6c shows gene prediction specificity for each methods' aggregation using the intersection operator. The methods' aggregations using the intersection operator increase the specificity of the prediction when compared to single methods and the methods' aggregations using the union operator, as it is illustrated in Fig. 6c. However, there are several genes that are not predicted by any method (e.g., AC068580, AC079630, AC021607). Sensitivity, on the other hand, has a different behavior for each methods' aggregation (Fig. 6d). Some genes are more

Table 2 Ten best methods' aggregations

Methods' aggregations	Specificity	Sensitivity	Correlation coefficient	% Genes correctly predicted
Augustus ∪ GeneID	0.805	0.807	**0.791**	**80.00**
Augustus ∪ Genscan	0.771	0.839	0.786	74.17
Genscan ∩ GeneZilla	0.850	0.683	0.745	73.33
Augustus ∪ Genscan ∪ GeneID	0.757	0.851	0.782	71.67
Augustus ∪ Genscan ∪ GeneID ∪ GeneMark	0.725	0.864	0.767	70.83
Augustus ∪ GeneZilla ∪ GeneMark	0.706	0.874	0.760	70.00
Augustus ∪ Genzilla ∪ Genscan ∪ GeneID	0.700	**0.882**	0.756	69.17
Genzilla ∩ GeneID	**0.866**	0.612	0.706	68.33
Genzilla ∩ Genscan ∩ GeneID	0.711	0.870	0.758	67.50
Augustus ∩ Genzilla	0.851	0.640	0.714	66.67

Gene finding methods are ordered by descending number of genes correctly retrieved. A gene is considered correctly retrieved when its correlation coefficient is over 0.7. The best result for each column is highlighted in bold

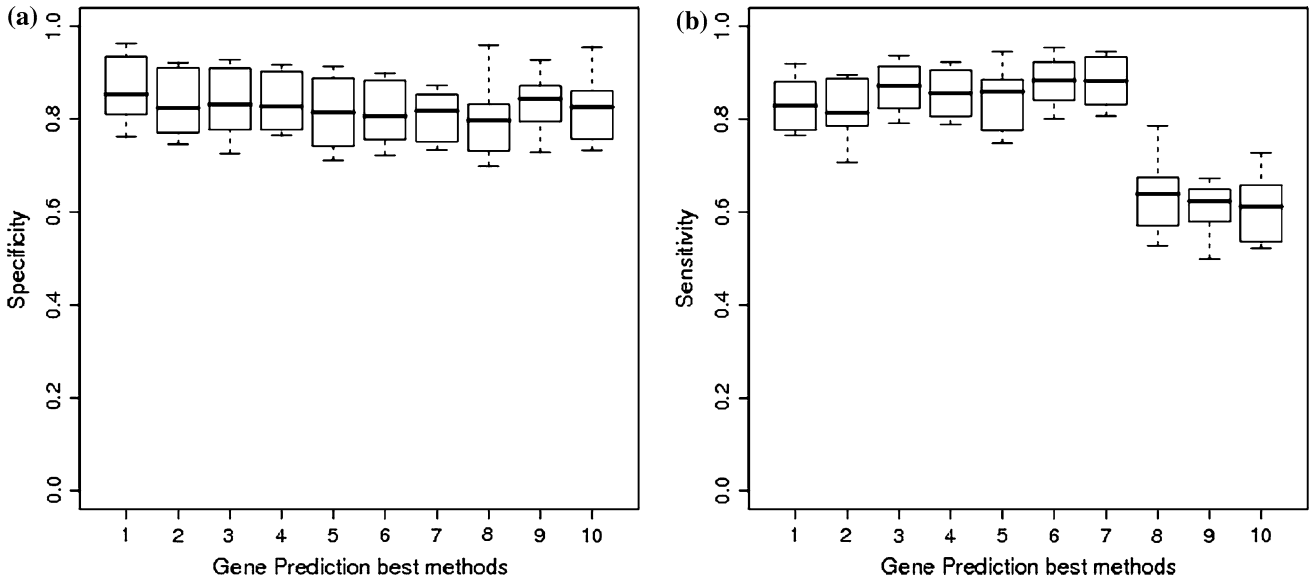

Fig. 5 Sub-sample boxplots. **a** Each boxplot represents the specificity obtained over all ten sub-sample sets from the original dataset with a method aggregation. **b** Each boxplot represents the sensitivity obtained over all sub-sample sets of the original dataset with a method aggregation. The methods' aggregations applied are the ones reported in Table 2 as the ten best methods' aggregation

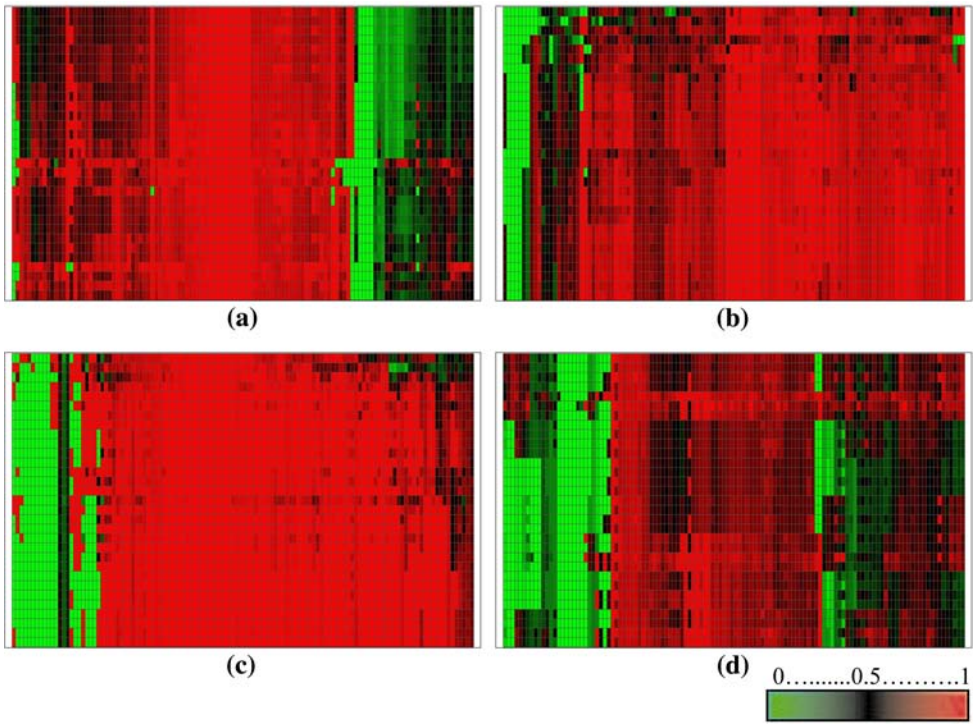

Fig. 6 Specificity and sensitivity of methods' aggregations obtained for the ENCODE dataset. Each *column* represents a gene of the ENCODE region (120 columns) and each *row* a method aggregation including the individual ones (31 rows). **a** Specificity for the union operator. **b** Sensitivity for the union operator. **c** Specificity for the intersection operator. **d** Sensitivity for the intersection operator. The *color* is coded from 0 (*green*) to 1 (*red*). *Green cells* represent low values while the *red cells* correspond to high values. *Black points* correspond to values around 0.5. Labels for the *x* and *y* axis are shown in additional Table 1

difficult to predict than others as represented by mostly red (e.g., AC072051, AL023881) or green columns (e.g., AC079630, AC068580). Finally, there are several genes which are not recognized by any method or methods' aggregations (e.g., AC068580, AC079630, AC021607).

3.2 Gene expression profile finding

The profiles conforming the dataset are the ones obtained from the inflammation and host response to injury problem. The methods analyzed in this study are: Student's *t* test

Table 3 Relabelling of methods analyzed in this study

Legend	Methods
M_1	Student's *t* test considering treatment versus control
M_2	Student's *t* test considering time
M_3	Permutation test considering treatment versus control
M_4	Permutation test considering time
M_5	ANOVA considering treatment versus control
M_6	ANOVA considering time
M_7	ANOVA considering treatment and time
M_8	RMANOVA considering treatment
M_9	RMANOVA over time
M_{10}	RMANOVA over treatment and time

Table 4 Results obtained by the individual methods, M_1 to M_{10}

Methods	Specificity	Sensitivity	Correlation Coefficient
M_1	0.553	0.560	**0.019**
M_2	0.266	0.776	0.007
M_3	**0.716**	*0.342*	0.011
M_4	0.502	0.551	0.009
M_5	*0.195*	**0.818**	0.003
M_6	0.477	0.530	0.002
M_7	0.346	0.656	*0.001*
M_8	0.455	0.546	*0.001*
M_9	0.559	0.451	0.002
M_{10}	0.621	0.392	0.002

The best result for each column is highlighted in bold, while the worst result is highlighted in italic

considering treatment versus control, Student's *t* test considering time, Permutation Test considering treatment versus control, Permutation Test considering time, ANOVA considering treatment versus control, ANOVA considering time, ANOVA considering treatment and time, RMANOVA considering treatment, RMANOVA over time, and RMANOVA over treatment and time. These methods have been relabelled for simplification purposes (Table 3). We show in Table 4 the average results to retrieve each of the 29 gene expression profiles obtained by the individual methods in terms of specificity, sensitivity and correlation coefficient (CC). Values are represented in the [0–1] interval.

Out of all gene expression methods analyzed, ANOVA considering time (represented by M_5) achieved the best sensitivity level and the Permutation Test considering time (M_3), the best specificity level. However, their average correlation coefficient (CC) for all profiles was not the highest. We can see that the levels of CC are generally low. This is due to the type of problem we are dealing with, finding a particular profile in a very large set of data, and

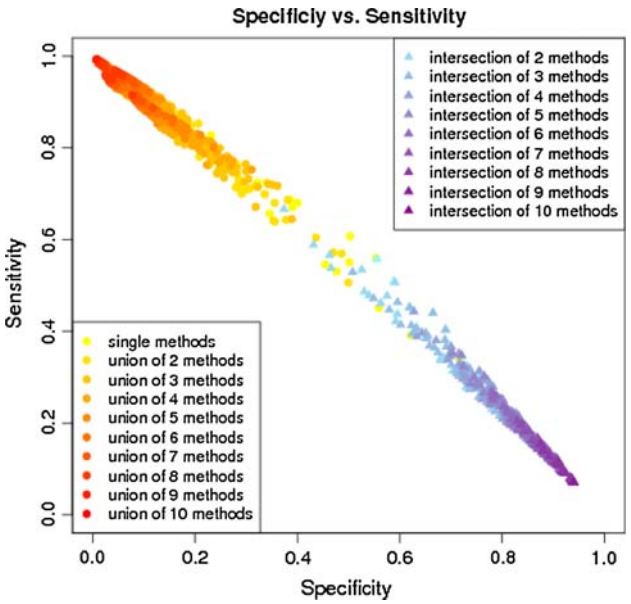

Fig. 7 Sensitivity versus specificity for all methods' aggregations and individual methods. The number of methods used in each aggregation is color-coded as indicated in the legend. The *light-yellow circles* represent sensitivity and specificity values of the individual methods

obtaining as a result a large rate of false positives (FP), which decrease dramatically the correlation coefficient associated to each method.

However, as occurred with the gene prediction problem, the results show that a low average CC does not imply a bad performance. Some methods recover particular profiles with better levels of CCs, and also with a good specificity/sensitivity aggregation levels.

All 1,023 potential methods' aggregations of the ten applied gene expression programs were evaluated. Figure 7 represents the general increase of sensitivity with the number of methods combined using the union operator, while specificity increases when combining with the intersection operator. Individual methods (in red), are in a middle position generally improved in terms of specificity/sensitivity by methods' aggregations. These methods' aggregations provide a wide spectrum of specificity/sensitivity levels depending on the operator applied. Union favors sensitivity, decreasing the number of false positives (FP), whereas intersection favors specificity, decreasing the number of false negatives (FN). The correlation coefficient is improved by application of any of the two operators.

We show in Table 5 the ten best methods' aggregations in terms of CC. We see how the CC levels widely overcome the levels obtained by the single methods. Methods aggregated by means of the union operator also highly improve the sensitivity levels, while methods obtained applying the intersection operator widely overcome the single ones in terms of specificity. The CC levels obtained

Table 5 Top ten methods' aggregations according CC obtained with the union and intersection operator

Methods' aggregation	Specificity	Sensitivity	Correlation coefficient
$M_1 \cup M_5 \cup M_7 \cup M_8$	0.076	0.912	**0.341**
$M_1 \cup M_3 \cup M_7 \cup M_8$	0.158	0.856	0.309
$M_1 \cup M_2 \cup M_3 \cup M_7 \cup M_8 \cup M_9$	0.074	**0.947**	0.261
$M_4 \cup M_5 \cup M_6 \cup M_7 \cup M_9 \cup M_{10}$	0.033	0.945	0.242
$M_1 \cap M_3 \cap M_9$	0.854	0.175	0.238
$M_1 \cap M_3 \cap M_5 \cap M_9$	0.854	0.175	0.238
$M_2 \cap M_4 \cap M_6 \cap M_7 \cap M_8 \cap M_9 \cap M_{10}$	**0.902**	0.113	0.235
$M_3 \cup M_5 \cup M_6 \cup M_{10}$	0.045	0.943	0.230
$M_4 \cup M_5 \cup M_7$	0.071	0.914	0.193
$M_1 \cup M_5 \cup M_7 \cup M_9 \cup M_{10}$	0.051	0.934	0.126

The values represent the average levels of specificity and sensitivity for the 29 gene expression profiles. The best result for each column is highlighted in bold

by the best ten methods range [0.235–0.341], while the individual correlation coefficient [0.002–0.019]. The sensitivity levels obtained by the methods' aggregations with the union operator range [0.856–0.947], while the sensitivity levels from the individual methods range [0.342–0.812]. Regarding the specificity, the intersection operator achieves levels ranging in the [0.854–0.902] interval, compared to the individual methods [0.195–0.716].

The best methods' aggregations applying the union operator include ANOVA considering time (M_5) and ANOVA considering time and treatment (M_7), which appear combined in four out of the seven best aggregations obtained with the union operator. In fact, the best aggregation in terms of correlation coefficient is $M_1 \cup M_5 \cup M_7 \cup M_8$, and when replacing M_5 by M_3 (Permutation Test considering time) the sensitivity value decreases from 0.912 to 0.856 with an increase in specificity from 0.076 to 0.158.

To compare the results of individual methods against the aggregation strategy we have represented in Fig. 8 individual methods in blue, methods' aggregations in black and the Pareto optimal front in red. We see how the methods' aggregations increase the results obtained by individual methods and are present in the optimal Pareto front.

To indirectly estimate the sampling bias of the methods' aggregations, we sub-sample the inflammation and host response to injury dataset in ten subsets without reposition. For each of the ten best methods' aggregations showed in Table 5 we calculate their specificity and sensitivity levels. The results obtained for each method aggregation over the ten subsets are represented with box plots in Fig. 9. We see a low level of variation in each method, which suggests the good performance and robustness of the methods' aggregations.

The levels of specificity and sensitivity obtained by each method's aggregation using the union and intersection operator to retrieve each of the 29 gene expression profiles are shown in Fig. 10. The rows and columns for each graph were clustered independently, and therefore we can see

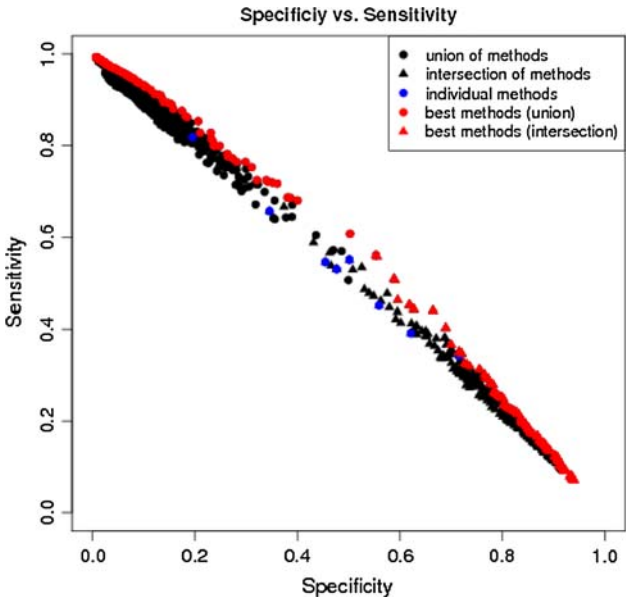

Fig. 8 Average specificity and sensitivity levels for the 29 gene expression profiles under study obtained by individual methods and the methods' aggregations. There are three color-codes: *blue dots* correspond to single methods, *red dots* to those methods' aggregations that are optimal in both objectives sensitivity and specificity (Pareto Optimal front); and *black dots* are all other methods' aggregations

groups (clusters) of method's aggregations showing similar behavior. The specificity of each method aggregation applying union to retrieve each gene expression profile is shown in Fig. 10a. We see low levels of specificity in general, since as we already stated, the union operator decreases dramatically specificity. There are some groups of rows with higher levels, black or dark red, which represent some methods' aggregations with better results for the specificity than the majority. The sensitivity of each methods' aggregation applying union to retrieve each gene expression profile is shown in Fig. 10b. In this case, the majority of the values are bright red, values close to 1, but there are certain profiles, the first two columns in

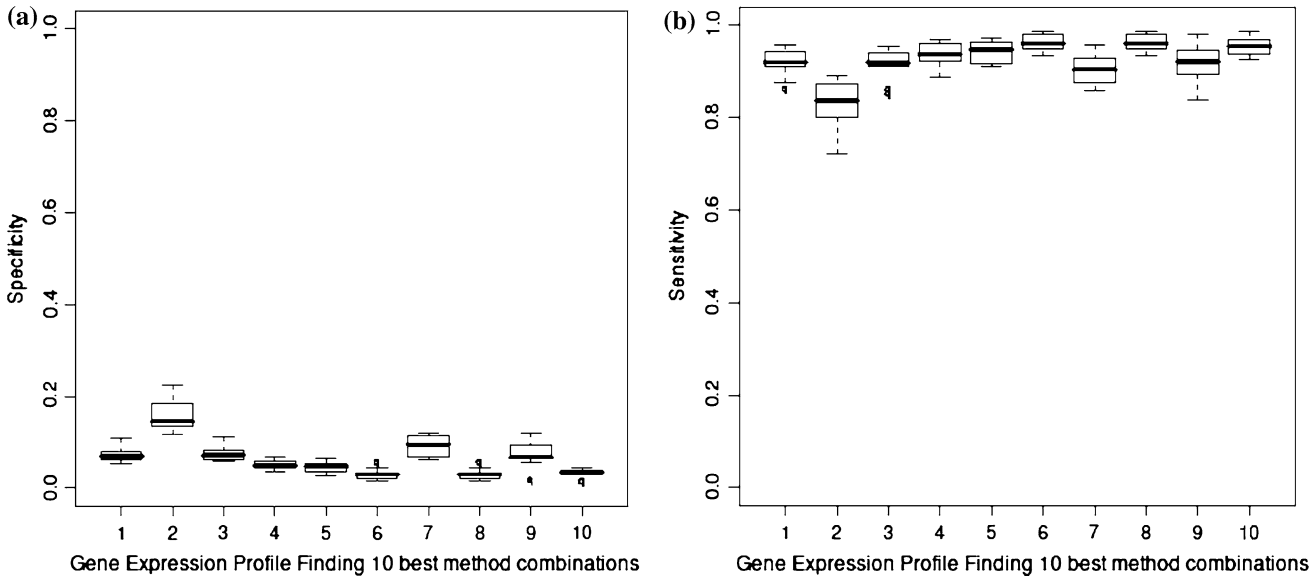

Fig. 9 Sub-sample boxplots. **a** Each boxplot represents the obtained specificity with a method aggregation over all ten sub-sample sets of the original dataset. **b** Each boxplot represents the obtained sensitivity with a method aggregation over all sub-sample sets of the original dataset. Methods' aggregations used are the same ones and in the same order as those reported in Table 5 as the ten best methods' aggregation

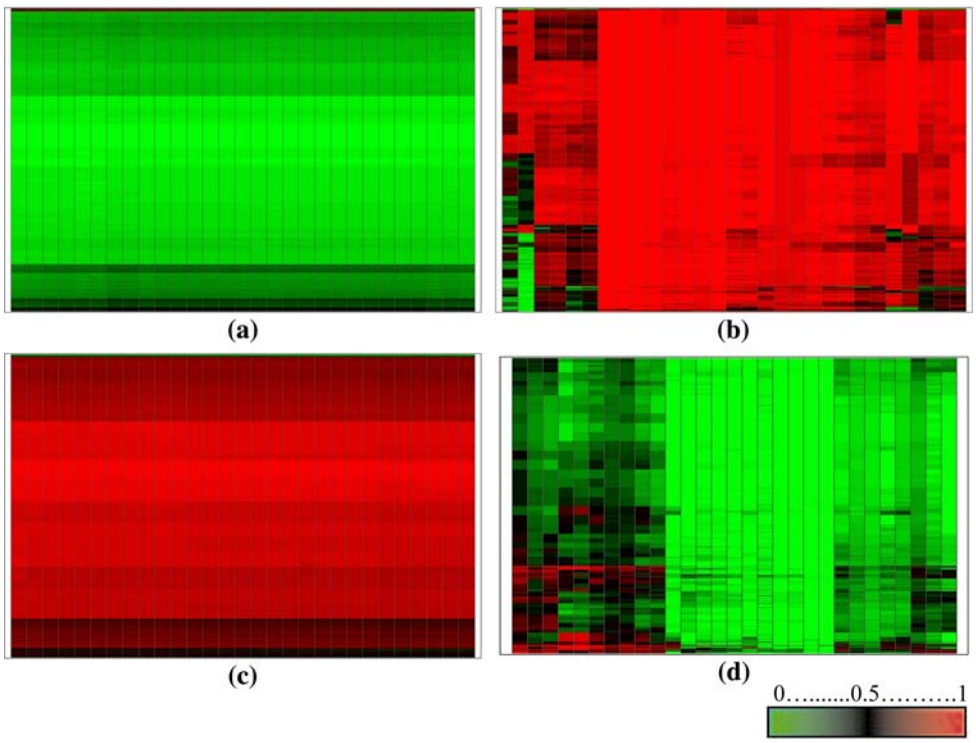

Fig. 10 Graphical representation of methods' aggregations specificity and sensitivity. Each *column* represents a gene expression profile (29 *columns*) and each *row* a method aggregation including individual methods (1,023). *Color* is coded from 0 (*green*) to 1 (*red*). Therefore, *green* cells represent low levels while *red points* correspond to high levels. *Black points* correspond to levels around 0.5. **a** Specificity for the union operator. **b** Sensitivity for the union operator. **c** Specificity for the intersection operator. **d** Sensitivity for the intersection operator. Labels for the *x* and *y* axis are shown in additional Table 2

particular, which are hard to recover by large groups of methods' aggregations. The isolated green cells represent profiles that are generally well recovered by the majority of methods, but show troubles with a particular aggregation. The specificity of each method aggregation applying intersection is shown in Fig. 10c. In this case we see very high values in general, showing a great capacity of methods' aggregations to be specific when intersected. The sensitivity levels of intersection, shown in Fig. 10d, confirm that intersection decreases sensitivity increasing the

rate of false negatives, although there are certain profiles with not so low levels, the profiles to the left of he figure.

4 Discussion

We propose a methodology to combine algorithms for a biological problem into an aggregation scheme. Our approach consists on the use of a multi-objective approach to extract the best methods' aggregation by maximizing the specificity and sensitivity of their predictions. This approach can provide better predictions by combining the advantages and strengths of the different algorithms available for a certain problem and avoiding redundant and overlapping predictions that might be produced depending on the methodologies and the aggregation scheme used.

The application of the proposed methodology to the gene finding and to the gene expression problem, shows in both issues a performance improvement of optimal methods' aggregation when compared to the individual methods for each topic.

When determining which methods' aggregation was the best one for the gene prediction problem, sensitivity and specificity were in contradiction. Nevertheless, the estimation of the correlation coefficient helped in the selection of the best methods' aggregations.

The best aggregations include methods employing different algorithmic strategies that predict correctly different subsets of the genes in the dataset. Although the statistical properties of coding regions allow for a good discrimination between large coding and non-coding regions, the exact identification of the limits of exons or of gene boundaries remains difficult. For instance, GeneID has strong constraints concerning this point. In case of alternative splicing, a predicted structure frequently splits a single true gene into several or, alternatively, merges several genes into one. Such problems are, however, very complex, as intergenic and intronic sequences do not differ much, and specific gene boundary signals in the UTRs (e.g. the TATA box and the polyadenylation signal), are often too variable and sometimes are not even present [37]. Some gene finders, like GeneZilla, obtain low specificity levels; this may be due to the fact that they were tested with unmasked sequences. It is well known that gene finding programs perform worse on unmasked sequences due to the high 'protein-coding-like' content of repetitive elements, resulting in an increase of the number of false positive predictions [38]. Augustus obtained very good results individually and takes part in many of the best methods' aggregations, showing robust results. Nevertheless, it was not able to identify some coding sequences that other gene finding methods could, such as Genscan and GeneMark for ENCODE region ENm011 and ENr322. The obtained results indicate that we could improve the exon accuracy by implementing a mixed approach doing the union only on the predicted regions of higher quality and doing the intersection for low-quality regions.

There are several previous publications combining gene finding programs [15, 39], but they fail to obtain good results as they use simultaneously all programs instead of optimizing their aggregation. De novo gene prediction for compact eukaryotic genomes is already quite accurate, although mammalian gene prediction lags way behind in accuracy. One future scope would be the extension of the application of this approach to identify ways to quickly combine many or all existing programs trained for the same organism, and determine the upper limit of predictive power by aggregations of programs genome wide [40].

The application of our methodology to standard analytical methods used for microarray experiments analysis alleviated the problems exhibited by individual methods, including missing important probe sets. The improvement in sensitivity was greater than 20% without a reduction of the specificity for the methods' aggregations used. Our approach was able to detect probe sets not reported in the first publication of the dataset [21], where two classic microarray analysis methods, M_1 and M_3 were individually applied. In fact, some of these probe sets have been shown to be related both in expression level and functionality to probe sets stated as relevant in the publication [21]. Such is the case of probe set *206011_at*, related to gene *CASP1*, found by applying our methodology [34], which is related in gene expression level (see additional Fig. 1) and in function (apoptosis-related cysteine peptidase) to probe sets *211367_s_at*, stated as relevant for the inflammation problem in [21]. Probe set *206011_at* was found by the method aggregation $M_7 \cup M_{10}$.

As well as in the gene finding problem, the aggregations of the different programs/methods resulted optimal and consistently outperformed even the best individual approach and, in some cases, produced dramatic improvements in sensitivity and specificity. Moreover, we observed that even the worst methods contributed to the aggregation with more accurate programs.

The proposed methodology applied to the microarray technology is valid for either providing the optimal methods' aggregations for a query profiles, or for identifying all differential profiles in a given set of microarray data suggesting the optimal methods' aggregations for them. Although we have applied our procedure to time-course structured experiments, they constitute a more general case of simpler microarray problems where microarray samples are taken as single data points. Therefore, the methodology presented is also useful for simpler microarray experiments with single data points.

Our approach presents various advantages over the standard analytical methods for microarray experiments. The aggregation of the union and intersection operators provides the possibility of querying negative samples (i.e., genes which exhibit a given profiles but not others). The representation used for the profiles is optimal, and allows us to examine the behavior of the genes independently in each subject, and facilitates the identification of different behaviors of genes across the subjects in the same experimental group. These differences can help us to discover the influence of biological conditions not previously considered in the experiment such as gender or age. In contrast to other approaches, the system provides solutions based on a trade-off of specificity versus sensitivity, whereas other methods evaluate their solutions over one measure, usually a ratio between false positives and the total number of genes retrieved. The computational procedure presented can solve some of the problems actually present in the process of analyzing a microarray experiment, such as the decision of analytical methodology to follow, extraction of biologically significant results, proper management of complex experiments harboring experimental conditions, time-series and inter-subject variation [34]. Therefore, it provides a robust platform for the analysis of many types of microarray experiments, from the simplest experimental design to the most complex, providing accurate and reliable results.

In the last 10 years, the existing competitive spirit has increased the number of programs/algorithms created, updated and adapted for the two biological problems here presented [1, 2, 4, 10, 28, 41]. On the one side, the development of a new algorithm always implies the sacrifice of an objective in favor of another, which makes very difficult for novel approaches to improve in absolute terms the quality of the existing ones. On the other side, the impressive amount of alternative algorithms available for different biological problems is confusing for users, who wonder what makes the programs different, which one should be used in which situation and which level of prediction confidence to expect. Finally, users also wonder whether current programs can answer all their questions. The answer is most probably no, and will remain to be negative as it is unrealistic to imagine that such complex biological processes can be explained merely by looking at one objective.

Our future work will extend the methodology here proposed in an automatic method generator, and a step forward to exploit all already existing methods, by providing optimal methods' aggregations to answer concrete queries for a certain biological problem with a maximized accuracy of the prediction.

Acknowledgments This work was supported in part by the Spanish Ministry of Science and Technology (MEC) under project TIN-2006-12879 and the Consejeria de Innovacion, Investigacion y Ciencia de la Junta de Andalucia under project TIC-02788. I. Zwir is a senior research scientist supported by the Howard Hughes Medical Institute and the "Ramon y Cajal" program of the MEC, C. del Val was supported by the "Programa de Retorno de Investigadores" from the Junta de Andalucia.

References

1. Mathé C, Sagot MF, Schiex T, Rouzé P (2002) Nucleic Acids Res 30:4103
2. Claverie JM (1997) Hum Mol Genet 6:1735
3. Guigó R (1997) Comput Chem 21:215
4. Haussler D (1998) Computational genefinding. Trends Biochem Sci Suppl:12–15
5. Smyth GK, Yang YH (2003) Methods Mol Biol 224:111
6. Inza I, Larranaga P, Blanco R, Cerrolaza AJ (2004) Artif Intell Med 31:91
7. Li C, Wong WH (2003) The analysis of gene expression data: methods and software, Springer, New York, pp 120–141
8. Pan W, Lin J, Le C (2001) Funct Integr Genomics 3:117
9. Park T, Yi SG, Lee S, Lee SY, Yoo DH, Ahn JI, Lee YS (2003) Bioinformatics 19:694
10. Tusher VG, Tibshirani R, Chu G (2001) Proc Natl Acad Sci USA 98:5116
11. Vaquerizas JM, Conde L, Yankilevich P, Cabezon A, Minguez P, Diaz-Uriarte R, Al-Shahrour F, Herrero J, Dopazo J (2005) Nucleic Acids Res 33:616
12. Liu DK, Yao B, Fayz B, Womble DD, Krawetz SA (2004) Mol Biotechnol 26:225
13. Liu Q, Mackey AJ, Roos DS, Pereira FC (2008) Bioinformatics 24:597
14. Liu Q, Crammer K, Pereira FC, Roos DS (2008) BMC Bioinformatics 9:433
15. Murakami K, Takagi T (1998) Bioinformatics 14:665
16. Rhodes DR, Barrette TR, Rubin MA, Ghosh D, Chinnaiyan AM (2002) Cancer Res 62:4427
17. Grützmann R, Boriss H, Ammerpohl O, Lüttges J, Kalthoff H, Schackert HK, Klöppel G, Saeger HD, Pilarsky C (2005) Oncogene 24:5079
18. Guigó R, Reese M (2005) Nat Methods 2:575
19. Guigó R, Flicek P, Abril JF, Reymond A, Lagarde J, Denoeud F, Antonarakis S, Ashburner M, Bajic VB, Birney E, Castelo R, Eyras E, Ucla C, Gingeras TR, Harrow J, Hubbard T, Lewis SE, Reese MG (2006) Genome Biol 7:S2
20. ENCODE Project Consortium (2004) Science 306:636
21. Calvano SE, Xiao W, Richards DR, Felciano RM, Baker HV, Cho RJ, Chen RO, Brownstein BH, Cobb JP, Tschoeke SK, Miller-Graziano C, Moldawer LL, Mindrinos MN, Davis RW, Tompkin RG, Lowry SF, Inflammation and Host Response to Injury Large Scale Collab. Res. Program (2005) Nature 437:1032
22. Halmos P (1960) Naïve set theory, Princeton, NJ
23. Mitchell TM (1997) Machine learning, McGraw-Hill, New York
24. Cohon JL (1978) Multiobjective programming and planning, Academic Press, New York
25. Bauer E, Kohavi R (1999) Mach Learn 36:105
26. ENCODE Project Consortium (2007) Nature 447:799

27. Guigó R, Knudsen S, Drake N, Smith T (1992) J Mol Biol 226:141
28. Burge C, Karlin S (1998) Curr Opin Struct Biol 8:346
29. Borodovsky M, Lomsadze A, Nikolai I, Ryan M (2003) Curr Protoc Bioinformatics, Chap 4, Unit 4.6
30. Stanke M, Morgenstern B (2005) Nucl Acids Res 33:W465
31. Majoros WH, Pertea M, Salzberg SL (2004) Bioinformatics 20:2878
32. Burge C, Karlin S (1997) J Mol Biol 268:78
33. Burset M, Guigó R (1996) Genomics 34: 353
34. Rubio-Escudero C (2007) Fusion of knowledge towards the identification of genetic profiles in the systemic inflammation problem, University of Granada
35. Everitt B, Der G (1996) Statistical analysis of medical data using SAS, Chapman & Hall, London
36. Chankong V, Haimes YY (1983) Multiobjective decision making: theory and methodology, North-Holland, Amsterdam
37. Lim LP, Burge CB (2001) Proc Natl Acad Sci USA 98:11193
38. Bedell JA, Korf I, Gish W (2000) Bioinformatics 16:1040
39. Tech M, Merkl R (2003) In Silico Biol 3:441
40. ENCODE Project (2007) Nature 447:799
41. Li C, Wong WH (2001) Genome Biol 2:193